

# Lecture 7:

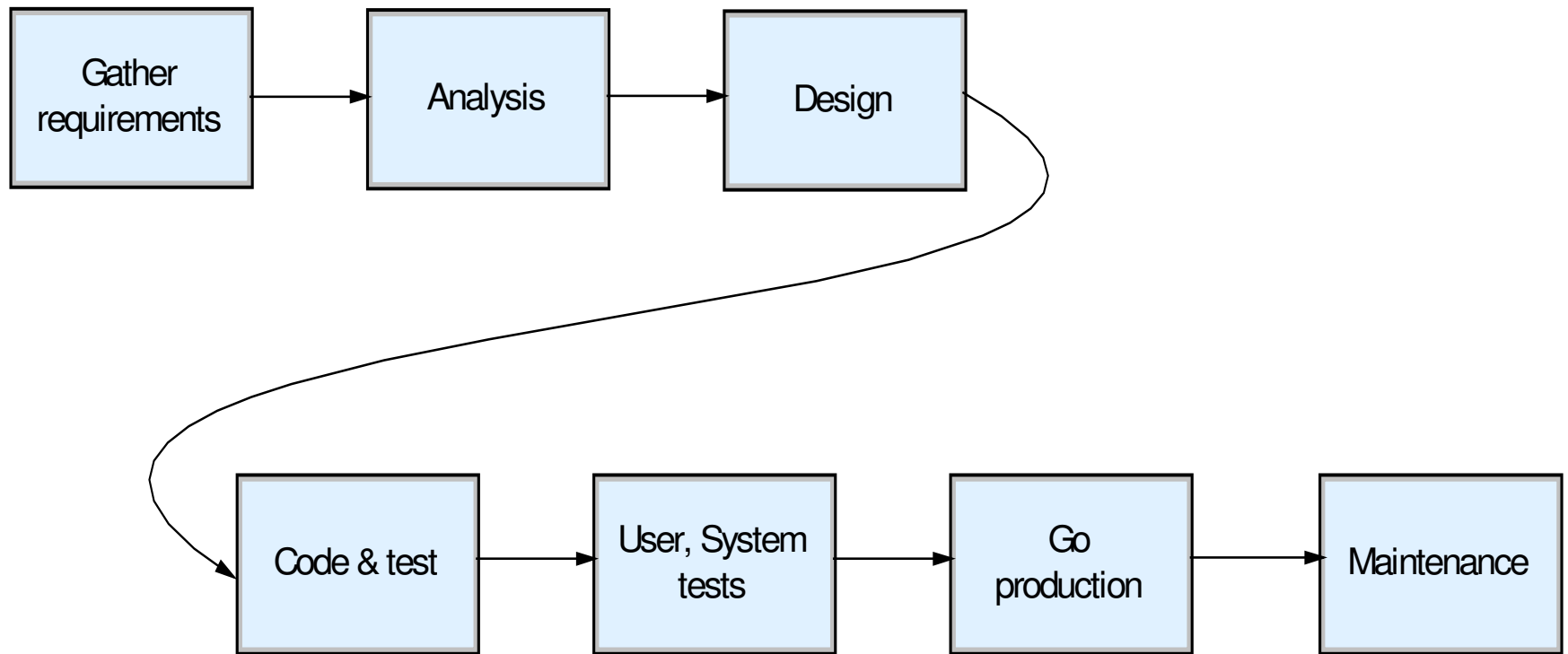
# Creating Large Scale Applications

# Reviewing the Software Development Life Cycle

# Overview of Development Life Cycle

- Designing and developing an application for the mainframe is similar to other platforms, but some of the questions and conclusions are different.
- Life cycle of designing and developing an application to run on z/OS includes phases of:
  - Requirements gathering and analysis
  - Design
  - Development
  - Test and debugging
  - Production
  - Maintenance

# Application Development Lifecycle



# What Percentage of Time is Typically Spent “Coding”?

Analysis	Design	Build	Test

# What Percentage of Time is Typically Spent “Coding”?

Analysis	Design	Build	Test
15%	30%	30%	25%

# Gathering Requirements for the Design

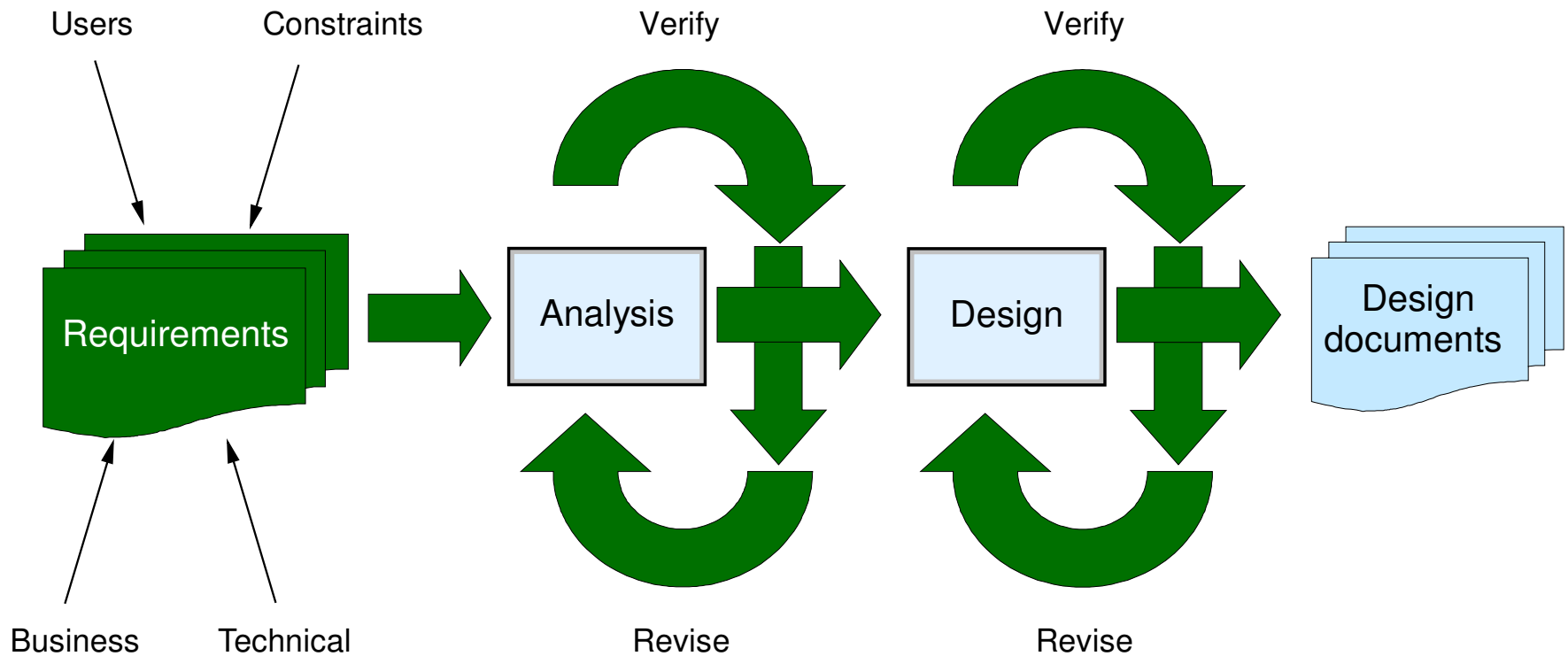
- Requirements:
  - Assess what needs to be accomplished
    - Based on projects constraints
    - Always keep in mind the end result
    - Conduct interviews with users and stakeholders
    - State and verify our assumptions

# Types of Requirements

- Accessibility
- Client
- Interoperability
- Recoverability
- Serviceability
- Availability
- Connectivity
- Performance
- Portability
- Usability
- Frequency of data backup
- Distributed
- Secure centralized controllable capacity
- Web services
- Changeability
- Preventing failure and fault analysis
- Resource can be monitored, controlled, managed, and administered



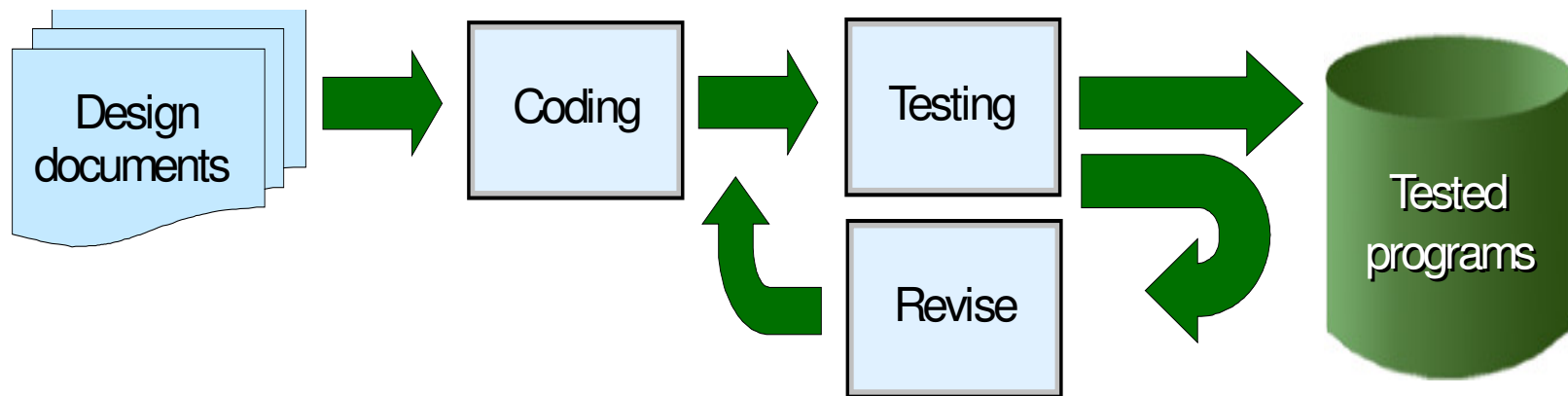
# Design Phase



# Design Decisions – *based on requirements*

- Batch versus online
- Database, tape, flat file, etc.
- Capacity of server
- Server type
- Develop or purchase package or both

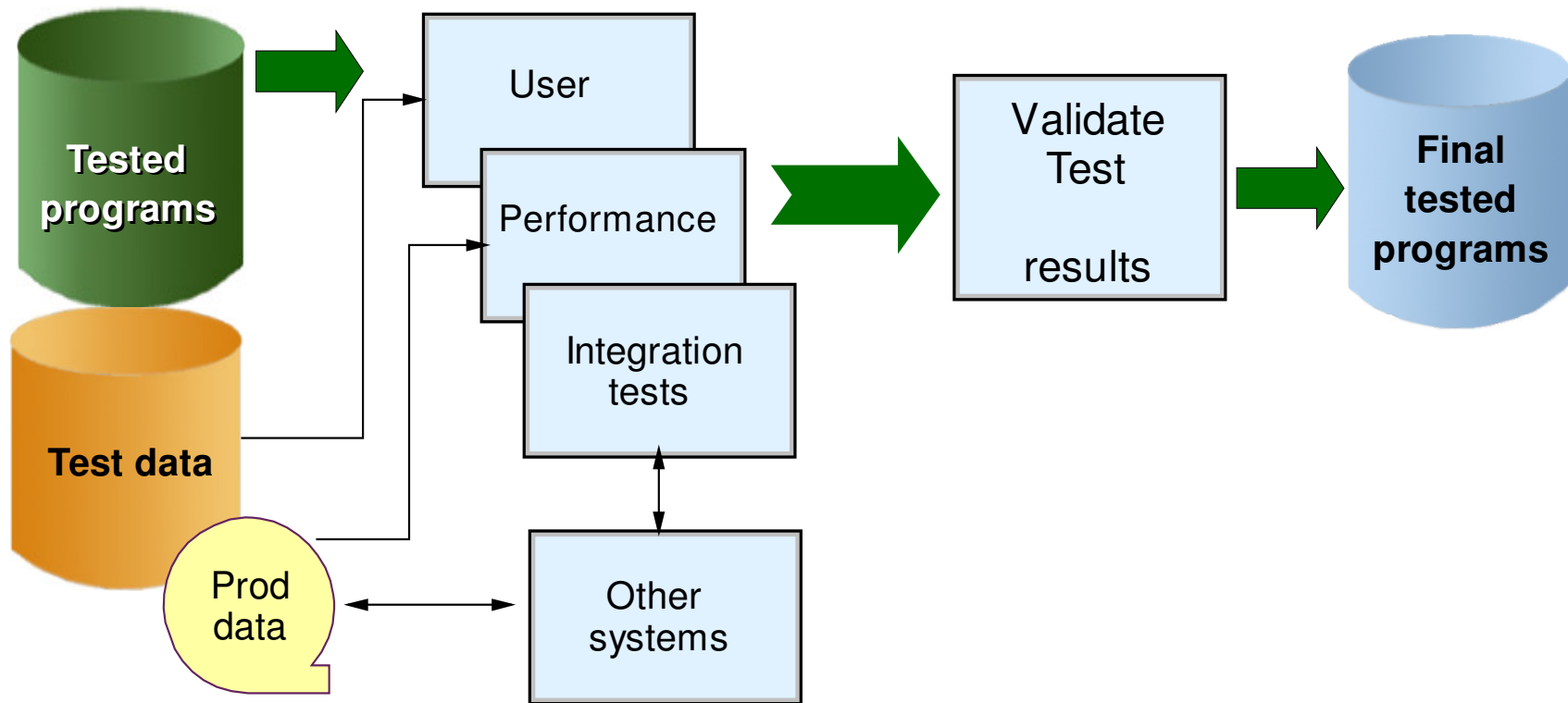
# Development Phase



# Developing an Application for the Mainframe

- Programmer uses as input the specifications of the designer
- Usually follows this process:
  - Code a module.
  - Test a module for functionality.
  - Make corrections to the module.
  - Repeat from step 2 until successful.

# Test Phase



# Test Phase Questions

- How to do performance testing if you can only afford one system, and it's in production?

# Test Phase Questions

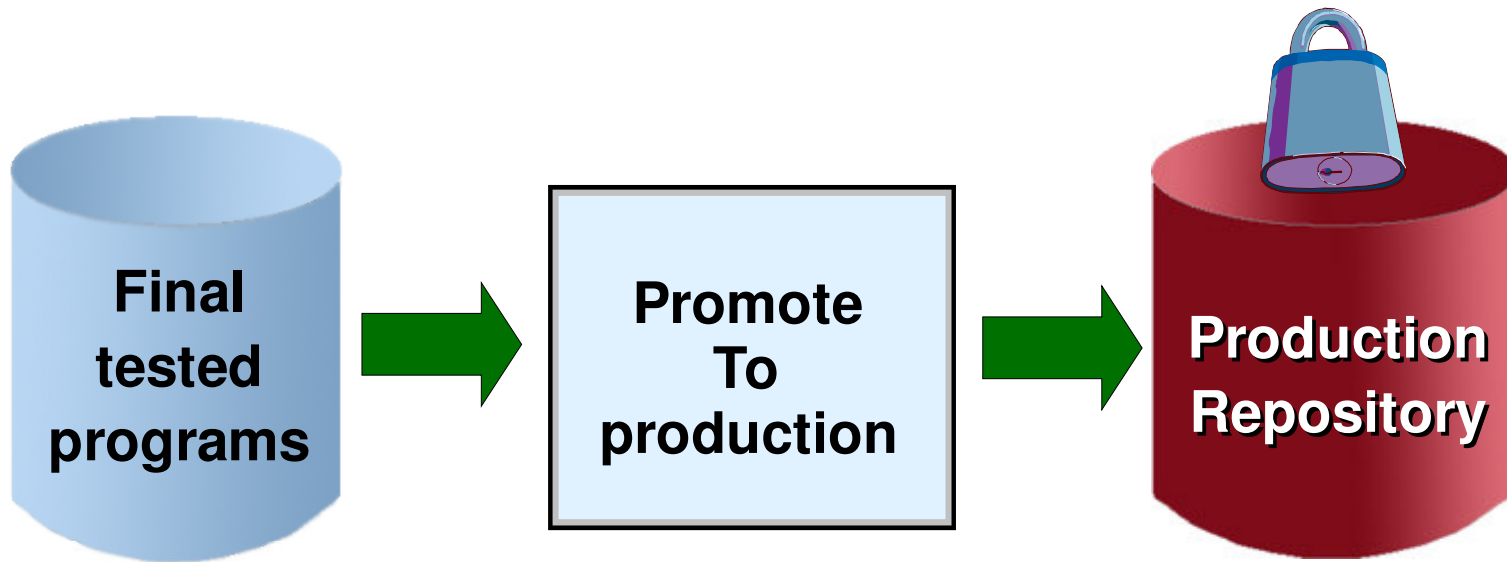
- How to do performance testing if you can only afford one system, and it's in production?
- Why is it better to separate the role of the tester and the role of the developer

# Test Phase (continued)

- Many levels of testing
  - User testing for functionality, acceptance
  - Performance (stress) testing
  - Integration testing (with other systems)
- Validate the testing results
- Final step before going production



# Production Phase



# Go Production

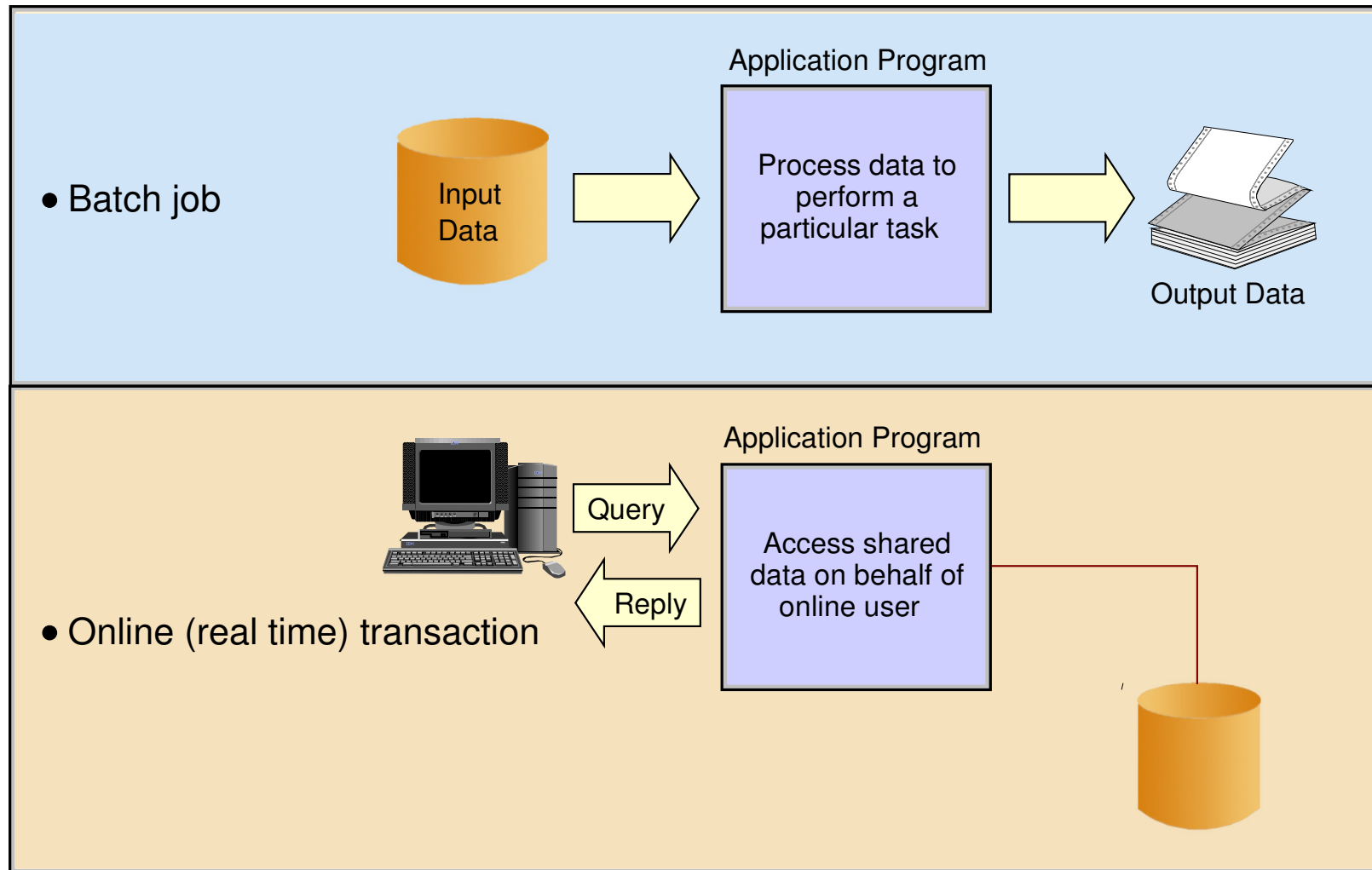
- Document:
  - Operational procedures
  - Training manuals (users, administrators, etc.)
- Promote application to production status
  - Implement change control process
- Hand over to operations

# Maintenance Phase

- Ongoing day-to-day changes/enhancements
- Responsibility for maintenance may change to another group or stay with developers

# Types of Applications: Batch & Online Processing

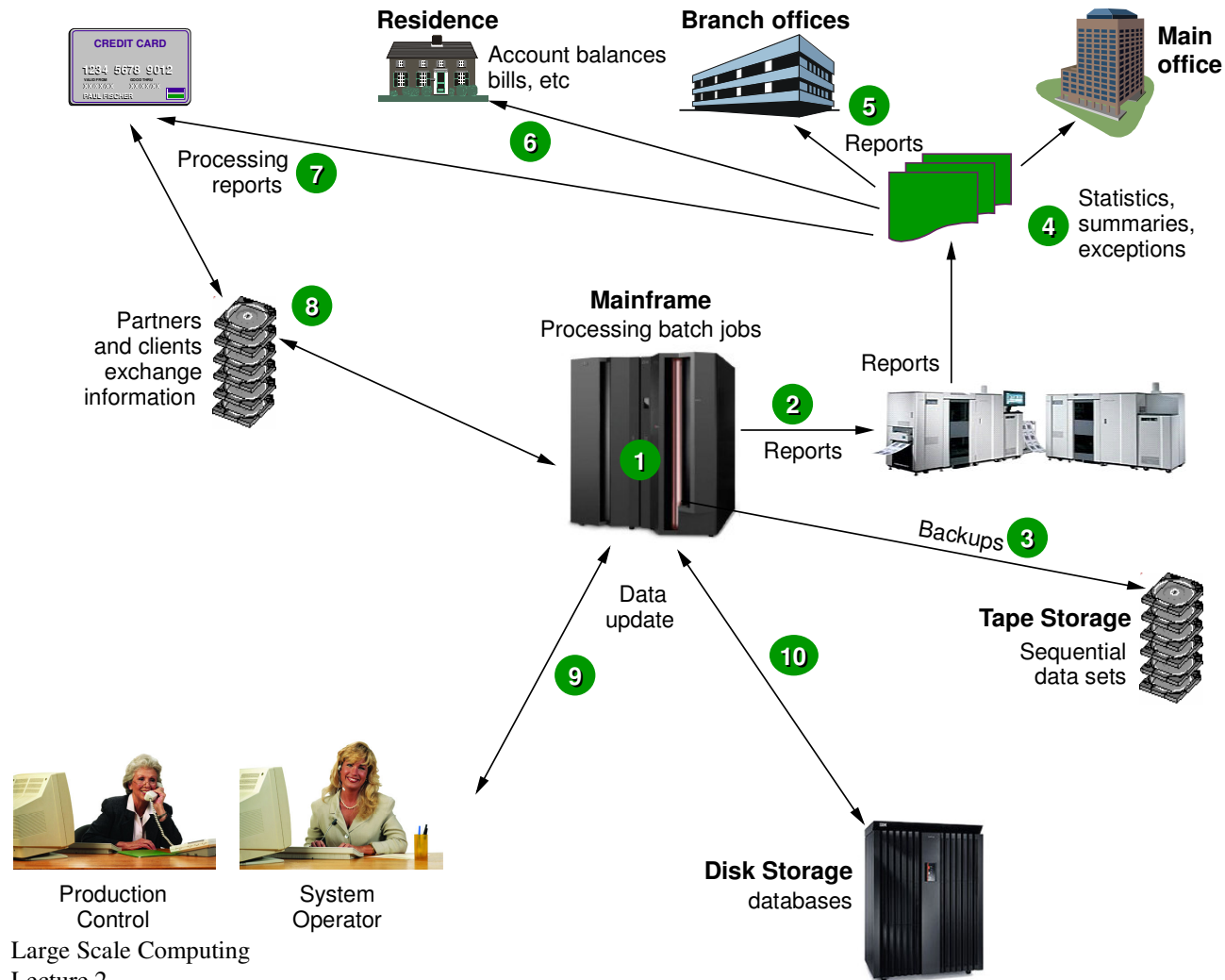
# Typical mainframe workloads



# What is batch processing?

- Batch processing is used for programs that can be executed:
  - With minimal human interaction
  - At a scheduled time or on an as-needed basis.
- After a batch job is submitted to the system for execution, there is normally no further human interaction with the job until it is complete.

# Typical batch use



# Examples of batch processing:



# Examples of batch processing:

- New Credit Card Applications

# Examples of batch processing:

- New Credit Card Applications
- Payments on credit cards

# Examples of batch processing:

- New Credit Card Applications
- Payments on credit cards
- Credit Card Statements / Interest rate calculations

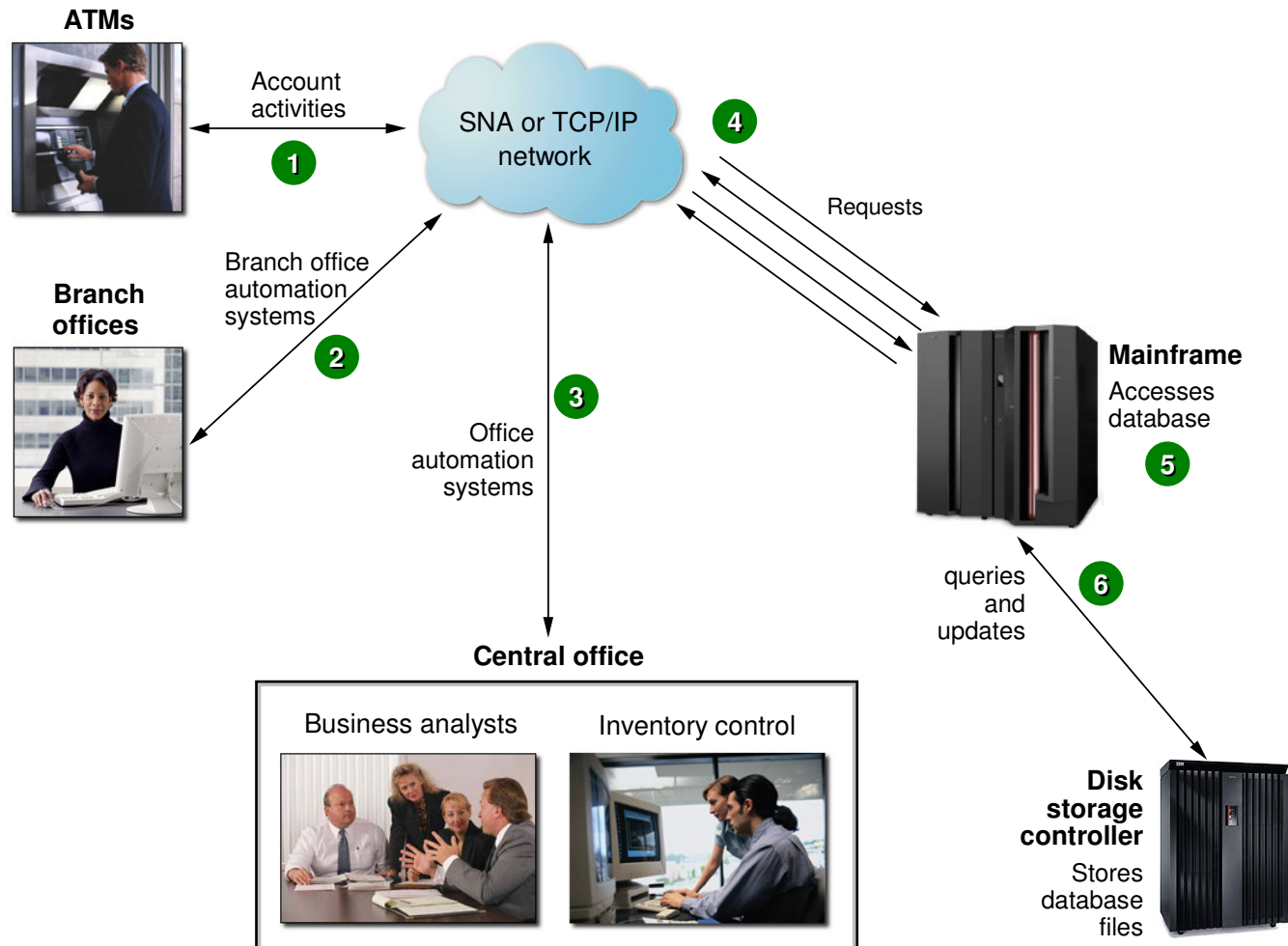
# Example of online processing: a travel agency

- Mainframe applications designed for:
  - employee and customer information
  - contacts with car rental companies
  - hotels
  - airline schedules
- Changes must be immediately reflected to application end-users (in real time)
- Contrast with batch processing

# What is online processing?

- Online processing is used for programs that require:
  - Human interaction
  - Can not be a scheduled

# Typical online use



# When to use which Batch vs Online Processing

- Reasons for using batch:
  - Data is stored on tape
  - Transactions are submitted for overnight processing
  - User does not require access to the data
- Reasons for using online:
  - Users require access to the data
  - Short response time required

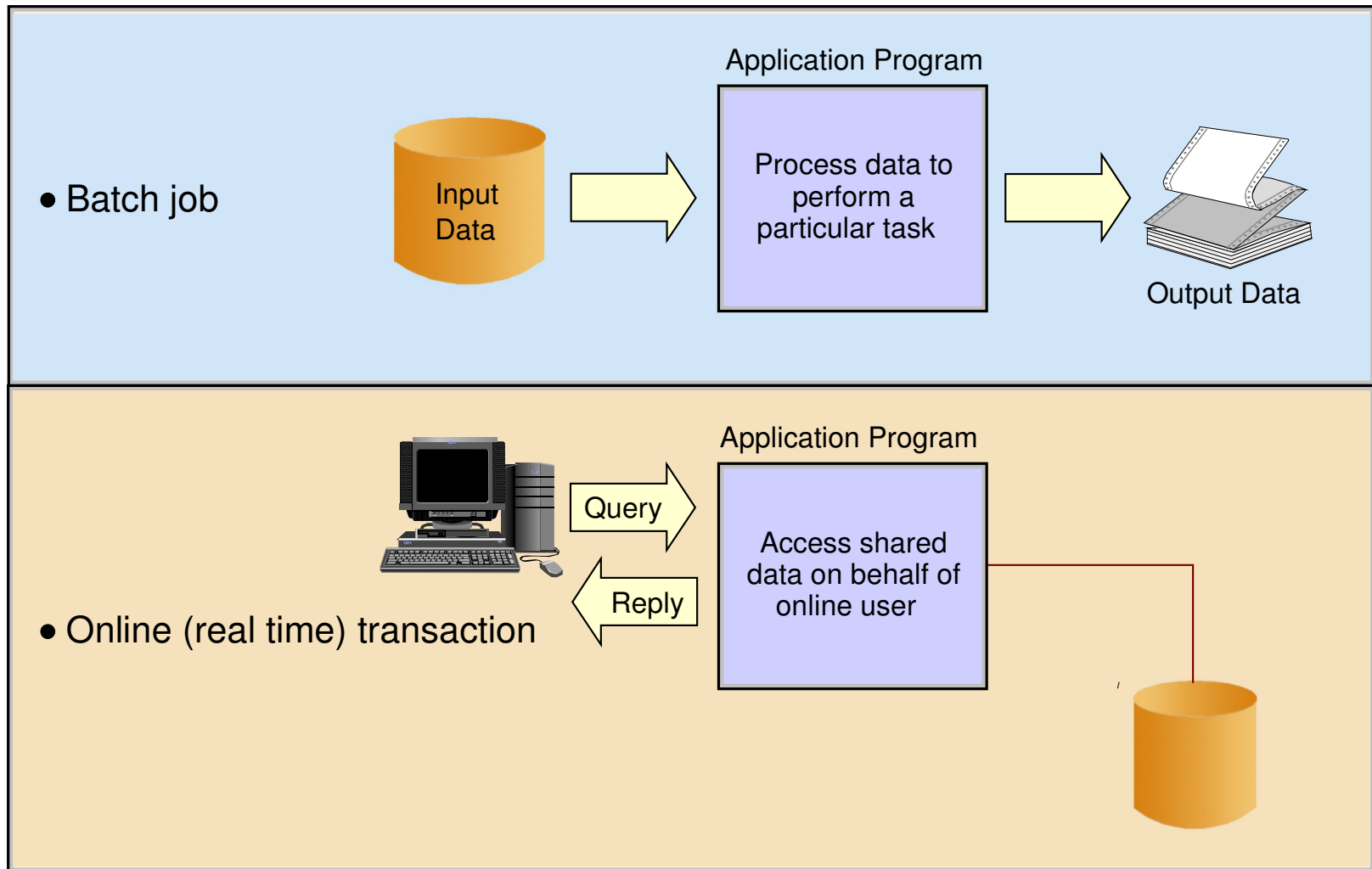
# Our CPS project (Card Processing System)



# Credit Card Transaction Processing Requirements

- Supports both real-time and batch processing
- Needs to be implemented in a modular fashion
- Key “real-time” processes:
  - authorization requests
- Key batch processes:
  - posting of payments
  - statements

# Reviewing online & Batch



# Nightly Batch Processing

- **Create accounts**
  - credit limit, daily “open to buy”
  - Customer info (name, address, phone, ...)
  - Bill date, payment due date
  - Can assume all payment due on “payment due date”
- **Update open to buy**
  - includes payments posted that day
  - Based on open-to-buy and other key metrics, determine if authorizations are approved.
- **Generate a statement file (that could be printed).**
  - This occurs once per month for each account.
  - Lists transactions, current credit limit and amount due

# Real-Time Authorizations

- Keep track of “open to buy” -- a variable that defines how much a person can spend in 1 day
- Based on open-to-buy and other key metrics, determine if authorization is approved.
- Reduce open to buy based on authorization amount (if approved)

# Customer Service

- **Support customer “call centers” answer:**
  - what is my current credit limit
  - how much do I have available to spend
  - what is my current bill

# Sort & Merge

# Sorting

- The StudentFile is a sequential file sequenced upon ascending StudentId.
- Write a program to display the number of students taking each course. How?

```
DATA DIVISION.  
FILE SECTION.  
FD StudentFile.  
01 StudentDetails.  
    02 StudentId          PIC 9(7) .  
    02 StudentName.  
        03 Surname        PIC X(8) .  
        03 Initials       PIC XX.  
    02 DateOfBirth.  
        03 YOBirth        PIC 9(2) .  
        03 MOBirth        PIC 9(2) .  
        03 DOBirth        PIC 9(2) .  
    02 CourseCode        PIC X(4) .  
    02 Grant              PIC 9(4) .  
    02 Gender             PIC X.
```

# Sorting - using COBOL

```
ENVIRONMENT DIVISION.  
INPUT-OUTPUT SECTION.  
FILE-CONTROL.
```

```
    SELECT WorkFile ASSIGN TO "WORK.TMP".
```

```
SD  WorkFile.  
01  WorkRecord.  
    02  ProvinceCode          PIC 9.  
    02  SalesmanCode          PIC 9(5) .  
    02  FILLER                PIC X(19) .
```

```
PROCEDURE DIVISION.
```

```
Begin.
```

```
    SORT WorkFile ON ASCENDING KEY ProvinceCode  
                  DESCENDING KEY SalesmanCode  
    USING UnsortedSales  
    GIVING SortedSales.
```

```
    OPEN INPUT SortedSales.
```



# Sorting - using JCL

- But most people do not use Cobol **internal** sorting because you can also sort using JCL

```
//JOBSORT JOB
//STEP1 EXEC PGM=ICETOOL
//IN          DD DSN='KCXXXX.LAB2.NEWACCTS',DISP=SHR
//SYSOUT      DD *
//TOOLMSG     DD SYSOUT=*
//DFSMSG      DD SYSOUT=*
//OUT         DD DSN=KCXXXX.LAB2.NEWSORT,DISP=SHR
//TOOLIN      DD *
              SELECT FROM(IN) TO(OUT) ON(1,10,CH) LAST
/*
/*
```

*For duplicates, can have: LAST, FIRST, NODUPS, ALLDUPS*

# MERGE Description

- The Merge takes two or more identically sequenced files and combines them, according to the key values specified, to produce a combined file which is then output to an output file or OUTPUT PROCEDURE.
- This can also be done in COBOL or JCL

e.g.

```
MERGE WorkFile ON ASCENDING KEY StudentId  
      USING InsertionsFile, StudentFile  
      GIVING NewStudentFile.
```